Remarks

The present amendment responds to the Official Action dated June 13, 2007. A petition for a one month extension of time to respond and authorization to charge our credit card the large entity extension fee of $120 accompany this amendment. The Official Action rejected claims 1,2, 4-6, 16, and 17 under 35 U.S.C. §102(e) as being anticipated by Dowling U.S. Patent No. 6,823,505 (Dowling). Claims 3, 18, and 19 were rejected under 35 U.S.C. §103(a) based on Dowling in view of Nair et al. U.S. Patent No. 6,944,747 (Nair). Claims 7-10 were rejected under 35 U.S.C. §103(a) based on Pechanek et al. U.S. Patent No. 6,173,389 (Pechanek 6,173,389) in view of Dowling. Claims 11-14 were rejected under 35 U.S.C. §103(a) based on Dowling in view of Choquette et al. U.S. Patent Publication No. 2002/0199084 (Choquette). Claim 15 was rejected under 35 U.S.C. §103(a) based on Dowling in view of Choquette as applied to claim 14, and further in view of Nair. These grounds of rejection are addressed below.

Claims 1, 3, 11, 12, 16, 18, and 19 have been amended to be more clear and distinct. Claims 1-19 are presently pending.

Amendment to the Specification

The Abstract has been amended to removed the objected to phrase.

The Art Rejections

As addressed in greater detail below, Dowling, Nair, Pechanek 6,173,389, and Choquette do not support the Official Action's reading of them and the rejections based thereupon should

be reconsidered and withdrawn. Further, the Applicant does not acquiesce in the analysis of Dowling, Nair, Pechanek 6,173,389, and Choquette made by the Official Action and respectfully traverses the Official Action's analysis underlying its rejections.

The Official Action rejected claims 1, 2, 4-6, 16, and 17 under 35 U.S.C. §102(e) as being anticipated by Dowling. Dowling addresses programmable address modes for calculating effective addresses based on address values stored in address registers for operands stored in a data memory. Dowling, Abstract. Dowling's processor fetches a data element from a data memory 120 in response to an address generated in an address arithmetic unit, such as AAU 106 or programmable AAU 212 of Fig. 2, based on one of the address registers AR0, AR2, ... ARn of register set 102. Dowling, col. 7, lines 48-50 and col. 8, lines 35-48, for example. Dowling describes such effective address generation by a programmable AAU in more detail with regard to Figs. 3A and 3C. In Figs. 3A and 3C, it is shown that the programmable AAUs 301 and 350, respectively, both receive an address value as input from the address registers. Dowling further states that the programmable AAU 301 may be used "as the programmable AAU 212 in the processor 200 shown in Fig. 2" which "allows a programmer to programmably permute the bits in the address registers AR0-ARn in the register set 102." Dowling, Figs. 2, 3A, and 3C and col. 10, lines 14-18. Dowling merely extends the capabilities of an address arithmetic unit (AAU) by programmably modifying address values stored in address registers.

In addition, the Examiner suggests that Dowling discloses an instruction register for receiving an instruction encoded with an operand address and control information indicating the operand address is to be translated as part of the instruction's execution. The Examiner cites

table 5, col. 13, lines 19-35 for support. The Examiner further suggests that in response to the instruction received in the instruction register, the operand address is translated to form the different operand address. The Examiner cites the same table 5 and col. 13, lines 19-35 for support.

However, table 5 lists program code made up of load and store instructions with an operand addresses that specify address registers, such as address register 0 (AR0), address register 4 (AR4), and address register 5 (AR5). The load and store instruction encoding would require a bit field to specify these address registers. Dowling does not transform this bit field encoded with the selected address register, but rather operates to transform the contents of the selected address register, as noted above with regard to Figs. 3A and 3C. Therefore, Dowling does not "directly" translate "the operand address bit field received as input to form the different operand address" with "the different operand address accessing a data element from the memory through the address inputs" as claimed in amended claim 1. To sum up, Dowling does not teach and does not make obvious claim 1 as presently amended.

Regarding claim 4, the Examiner suggests that Dowling's Fig. 4A, column 30-65, discloses a block load instruction and further suggests that a block load instruction is a typical matrix operation instruction in processor TMS320C2x. However, Dowling's Fig. 4A merely illustrates a matrix having sixteen data elements. Since the Examiner did not indicate a specific col. reference, the lines 30-65 following the introduction of Fig. 4A in col. 10 are presumed to be the referenced text. The text at Dowling's col. 10, lines 30-60 does not indicate a block load instruction, but merely indicates a manner of storing the data elements of the matrix shown in

Fig. 4A in a linear memory and how individual data elements may be addressed. Dowling, col. 10, lines 54-60. Dowling does not teach and does not make obvious a block load instruction.

Regarding claim 16, the Examiner relies upon Dowling and the suggested rationale used in the rejection of claim 1 to reject claim 16. As noted above, the suggested rationale for rejecting claim 1 is not valid. Dowling does not teach and does not make obvious amended claim 16.

Claims 3, 18, and 19 were rejected under 35 U.S.C.§103(a) based on Dowling in view of Nair. Nair addresses a matrix data processor and "an indexing methodology 300" operating on data elements of various resolutions, such as 4-bit, 8-bit, 16-bit, and so forth, by selecting the data elements by using indices representing the data elements. Nair, col. 6, lines 29-48. The indexing methodology 300 includes a starting index of a first element of a first source matrix, a starting index for a first element of a second source matrix, a starting index for a first element of a destination matrix, and an interval between elements for selecting among successive elements of the matrices. Nair, col. 8, lines 47-50 and col. 10, line 65 – col. 11, line 15.

Regarding claim 3, the Examiner states that Dowling does not expressly disclose providing the bit complementing function for matrix operation corresponding to the claim's "e bits" function. The Examiner depends upon Nair to resolve this deficiency. The Examiner suggests that Nair describes an apparatus for matrix processing that can be incorporated into Dowling's system to manipulate address bits using matrix multiply, addition, subtractions, bit-reverse operations etc. and cites Nair column 12, table 1 for support. Nair, column 12, table 1, is a listing of specific matrix operations that can be performed on data elements of a matrix. Nair,

col. 12, lines 14-22. Nair does not describe operations that can be performed on an "operand address bit field input" as presently claimed in claim 3. In the Response to Argument section of the Official Action, the Examiner states that Nair does not expressly disclose values being used are address values. As noted above, Dowling does not teach and does not make obvious the processor address translation apparatus of claim 1 and further provides no motivation to take data elements, such as the data elements of Nair, and treat a data element as an "operand address bit field input". Nair does not teach and does not make obvious claim 3.

Amended claim 18 addresses an "address translation method for translating a first address of a first data element in an address translation memory to a second address of a second data element in the address translation memory." Claim 18 claims "loading the set of {s,e} bits into an address translation parameter control register located in the address translation memory" and "enabling an address translation unit located in the address translation memory for translation." Dowling's data memory 120 has an address input, a bidirectional data port, and a data memory storage. Dowling, Fig. 2 and col. 7, lines 56-59. Dowling's data memory 120 is not an "address translation memory" having an "address translation control register" or having an "address translation unit" as claimed in claim 18. Dowling's data memory does not have an "address translation parameter control register located in the address translation memory." Dowling also does not have "an address translation unit located in the address translation memory." Rather, Dowling's data memory 120 is a separate memory having no additional internal capabilities, such as controlled by an "address translation control register" or "an address translation unit" as claimed in claim 18. The Examiner suggests that Dowling's Fig. 2, operand address translation

function #212 translates addresses. Dowling's programmable address arithmetic unit 212 is not an address translation unit located in an address translation memory. As noted above, Dowling's data memory 120 does not contain Dowling's programmable address arithmetic unit 212. Dowling does not teach and does not make obvious an address translation method as claimed in claim 18.

Claims 7-10 were rejected under 35 U.S.C. §103(a) based on Pechanek 6,173,389 in view of Dowling. The Examiner states that Pechanek 6,173,389's Fig. 3 illustrates register file addressing/indexing logic corresponding to claim's RFI register file indexing logic and updating logic. However, Pechanek 6,173,389's Fig. 3 does not illustrate RFI register file indexing logic and updating logic, but merely illustrates a ManArray 2x2 iVLIW SIMD processor 100 which uses techniques for selecting one or more sub-instructions from a VLIW for execution. It is brought to the Examiner's attention that Barry et al. U.S. Patent No. 6,446,190 (Barry) having Gerald Pechanek as a co-inventor does describe methods and apparatus for register file indexing. It is noted that U.S. Patent No. 6,446, 190 is incorporated by reference in its entirety in the present specification at page 13, lines 19-22. However neither patent, 6,173,389 nor 6,446,190, describes an "address translation unit for accessing the memory in a translation pattern, receiving a sequence of operand addresses from the multiplexer and, in response to the sequence of RFI operand addresses, translating the sequence of RFI operand addresses to form a sequence of different operand addresses in accordance with the translation pattern, the different operand addresses each accessing a data element from the memory through the address input" as claimed in claim 7. See, Fig. 6 Rt address translator 634 and page 18, line 9 – page 19, line 3 of the

present specification. Neither Pechanek 6,173,389 nor Barry teaches or makes obvious the "address translation unit" as claimed in claim 7.

Claims 11-14 were rejected under 35 U.S.C. §103(a) based on Dowling in view of Choquette. Amended claim 11 addresses an "address translation memory device for accessing data at translated addresses." The Examiner suggests that Dowling's Fig. 2 processor address translation apparatus comprises #102 register file, #150 etc. programmable address translation logic, for translating an instruction #107 operand address to a different address and cites col. 13, Table 5 for support.

Dowling's register set 102 of address registers AR0, AR1, ..., ARn, the address arithmetic unit 106, the programmable address arithmetic unit 212, and the multiplexers 203 and 122 are all part of a function of Dowling's processor to execute instructions, such as load and store instructions. For example, Dowling's programmable AAU 212 takes an address value from a selected address register in register set 102 and manipulates the address value for storage back into an address register in register set 102 in response to a processor instruction. An address register from register set 102 may then be selected for addressing the data memory 120. Dowling's col. 13 Table 5 lists program code made up of load and store instructions with operand addresses that specify address registers, such as address register 0 (AR0), address register 4 (AR4), and address register 5 (AR5). Each of the load and store instructions specify an address register from the register set 102 AR0-ARn that are part of Dowling's processor. Dowling operates to manipulate the contents of an identified address register, as noted above with regard to Figs. 3A and 3C, separately from the data memory 120. In addition to Dowling's

register set 102 of address registers AR0, AR1, ..., ARn, the address arithmetic unit 106, the programmable address arithmetic unit 212, and the multiplexers 203 and 122 are also separate from the data memory 120 in order to carry out the execution of a processor instruction.

In contrast to Dowling, a storage unit 855 is shown in Fig. 8B of the present specification as having two address translator functions 860 and 865 located in the storage unit 855. See, for example, Fig. 8B and text at page 23, lines 1-11 of the present specification. Fig. 9 illustrates another example in which a two port memory unit 900 includes an address translator 940. See, for example, Fig. 9 and text at page 24, lines 10-21, of the present specification. Dowling does not teach and does not make obvious an "address translation unit located in the address translation memory device for accessing the storage device in a translation pattern, the address translation unit translating a first read address value coupled to the first read address input port in accordance with the translation pattern, to a second read address value coupled to the storage device second read address input port for reading data from the storage device at a translated address during a read operation" as claimed in claim 11. Emphasis added.

In a similar manner to claim 11, amended claim 12 claims an "address translation unit located in the address translation memory device, for accessing the storage device in a translation pattern, the address translation unit translating a first write address value coupled to the first write address input port in accordance with the translation pattern, to a second write address value coupled to the storage device second write address input port for writing data to the storage device at a translated address during a write operation." Dowling does not teach and does not make obvious claim 12.

Since dependent claims depend from and contain all the limitations of the independent claims, claims 2-6, 8-10, 12-15, 17, and 19 distinguish from the references in the same manner as their associated base claims and thus these claims are in order for allowance.

Conclusion

All of the presently pending claims, as amended, appearing to define over the applied references, withdrawal of the present rejection and prompt allowance are requested.

Respectfully submitted,

Peter H. Priest
Reg. No. 30,210
Priest & Goldstein, PLLC
5015 Southpark Drive, Suite 230
Durham, NC 27713-7736
(919) 806-1600